6.001 SICP

Orders of Growth

- Linear
- Exponential
- Logarithmic

Orders of growth of processes

- Suppose \( n \) is a parameter that measures the size of a problem
- Let \( R(n) \) be the amount of resources needed to compute a procedure of size \( n \).
- We say \( R(n) \) has order of growth \( \Theta(f(n)) \) if there are constants \( k_1 \) and \( k_2 \) such that \( k_1 f(n) \leq R(n) \leq k_2 f(n) \) for large \( n \).
- Two common resources are space, measured by the number of deferred operations, and time, measured by the number of primitive steps.

Order of growth examples

For each, find simplest and slowest growing \( f \) for which \( R(n)=\Theta(f(n)) \)

- \( R(n)=6 \)
  - \( \Theta(1) \)
  - \( 1 \cdot 1 \leq 6 \leq 6 \cdot 1 \) for all \( n \)
  - \( R(n)=n^2+3 \)
  - \( \Theta(n^2) \)
  - \( 1 \cdot n^2 \leq n^2+3 \leq 2 \cdot n^2 \) for all \( n > 2 \)
  - \( R(n)=6n^2+3n^2+7n+100 \)
  - \( \Theta(n^2) \)
  - \( 1 \cdot n^2 \leq 6n^2+3n^2+7n+100 \leq 7n^2 \) for all \( n > 100 \)
  - \( R(n)=2^n+7 \)
  - \( \Theta(2^n) \)
  - \( 1 \cdot 2^n \leq 2^n+7 \leq 2^n \cdot 8 \) for all \( n > 0 \)
  - \( R(n)=2^n+n^8 \)
  - \( \Theta(2^n) \)
  - \( 1 \cdot 2^n \leq 2^n+n^8 \leq 2^n \cdot 2n \) for all \( n > 64 \)

Examples

(defun fact (n)
  (if (= n 1)
      1
      (* n (fact (- n 1))))
)

(fact 4)
(+ 4 (fact 3))
(+ 4 (* 3 (fact 2)))
(+ 4 (* 3 (* 2 (fact 1))))
(+ 4 (* 3 (* 2 1)))
(+ 4 (* 3 2))
(+ 4 6)
24

Time = \( \Theta(n) \)
Space = \( \Theta(n) \)
Examples

(define (fact2 n)
  (define (helper cur k)
    (if (= k 1)
      cur
      (helper (* cur k) (- k 1)))
  )

  (helper 1 n))

(fact2 5)
(helper 1 5)
(helper 5 4)
(helper 20 3)
(helper 60 2)
(helper 120 1)

Time = \Theta(n) Space = \Theta(1)

Towers of Hanoi

• Three posts, and a set of different size disks
• any stack must be sorted in decreasing order from bottom to top
• the goal is to move the disks one at a time, while preserving these conditions, until the entire stack has moved from one post to another

Towers of Hanoi

(define move-tower
  (lambda (size from to extra)
    (cond ((= size 0) true)
          ((= size 1) (move-tower (- size 1) from to)
           (print-move from to)
           (move-tower (- size 1) extra to)))
  )

(define print-move
  (lambda (from to)
    (write-line `'Move top disk from `)
    (write-line from)
    (write-line `' to `)
    (write-line to)))

Orders of growth for towers of Hanoi

• Let t_b be the number of steps that we need to take to solve the case for n disks. Then
  \[ t_b = 2t_{b-1} + 1 = 2^2t_{b-2} + 2^1t_{b-1} + 1 = 2^n - 1 \]
• So in time we have \Theta(2^n) -- exponential
• In space, we have one deferred operation for each increment of the stack of disks -- \Theta(n) -- linear

Quiz

What is simplest expression for the order of growth of running time of procedure mul1 & mul2?

(define (mul1 n m)
  (define (help count ans)
    (if (= count 0)
      ans
      (help (- count 1) (+ m ans)))
  )

  (help n 0))

1. Theta(1)
2. Theta(2n)
3. Theta(n)
4. Theta(n2)